

Unraveling the Mind's Fractal Symphony: Module 10

Jos Prinsen, Caterina Ceccato, Anita Vrins, Ethel Pruss

2023-05-26

```
knitr::opts_chunk$set(echo = TRUE)
library(reticulate)  #(Ushey K, Allaire J, Tang Y 2023)
require(nonlinearTseries)  #(Garcia C. 2022)
require(tseriesChaos)  #(Antonio, Narzo FD. 2019)
library(ggplot2)  #(H. Wickham. 2016)
require(pracma)  #(Borchers H. 2022)

path_to_pk1A <- "C:\\Users\\Jos Prinsen\\Downloads\\RawEEGComplexSystemsRReadyAdaptive.pkl"
path_to_pk1R <- "C:\\Users\\Jos Prinsen\\Downloads\\RawEEGComplexSystemsRReadyRandom.pkl"
RawEEGComplexSystemsRReadyAdaptiveR <- py_load_object(path_to_pk1A)
RawEEGComplexSystemsRReadyRandomR <- py_load_object(path_to_pk1R)
```

Data description

The adaptive condition has 38 participants, whereas the Random condition has 37 participants. The study was within participants, meaning that each participant is both within the Adaptive and Random conditions. The random condition is missing Participant 30, due to issues in pre-processing.

Each participant has a number of events, which corresponds to a 10 second EEG segment within a robot storytelling task. The number of events is different based on the story that was told. For more information on the experiment please refer to the previous modules.

EEG data is Scale free

Traditional EEG analysis often makes the assumption that brain activity follows a stationary and Gaussian distribution. However, these assumptions simplify the complexity of brain dynamics. Research has shown that the PSD of EEG signals exhibits a power-law (1/f) behavior (Bénar et. al., 2019), implying scale-free dynamics (Bensal et. al., 2021). In other words, the power of the EEG signals at different frequencies follows a power-law distribution rather than a Gaussian distribution. Here we show the distribution of PSD of our EEG signals, to verify that it indeed exhibits power-law behavior.

```
 # Assume epochs_data is your NumPy array of shape (n_epochs, n_channels, n_samples)
epochs_list = RawEEGComplexSystemsRReadyAdaptive[0]

epochs_data = np.array([df.values.T for df in epochs_list])

sfreq = 256
ch_names = ["Fz", "C3", "Cz", "C4", "Pz", "P07", "Oz", "P08", "AdaptiveRandom", "FirstSecond", "Participi
ch_types = ['eeg', 'eeg', 'eeg', 'eeg', 'eeg', 'eeg', 'eeg', 'eeg', "misc", "misc", "misc"]
```

```
info = mne.create_info(ch_names=ch_names, sfreq=sfreq, ch_types=ch_types)
```

```
# Create the MNE EpochsArray object  
epochs = mne.EpochsArray(epochs_data, info)
```

```
from scipy.stats import linregress  
import mne  
power = epochs.compute_psd(method = "welch", fmin=1, fmax=30)
```

```
poweraverage = power.average(method="mean")  
  
# Extract the frequencies and power values  
frequencies = np.linspace(1.1, 30.0, num=233)  
power_values = poweraverage.get_data().mean(axis=0) # Average across channels  
  
# Convert to log-log scale  
log_frequencies = np.log10(frequencies)  
log_power_values = np.log10(power_values)  
  
# Perform linear regression on the log-log scale  
slope, intercept, r_value, p_value, _ = linregress(log_frequencies, log_power_values)
```

```
# Plot the power-law fit  
plt.loglog(frequencies, power_values, color='blue', lw=1.5, label='Power Spectrum')
```

```
plt.loglog(frequencies, 10 ** (intercept + slope * log_frequencies), color='red', linestyle='--', label='Power-law Fit')
```

```
plt.xlabel('Frequency (Hz)')
```

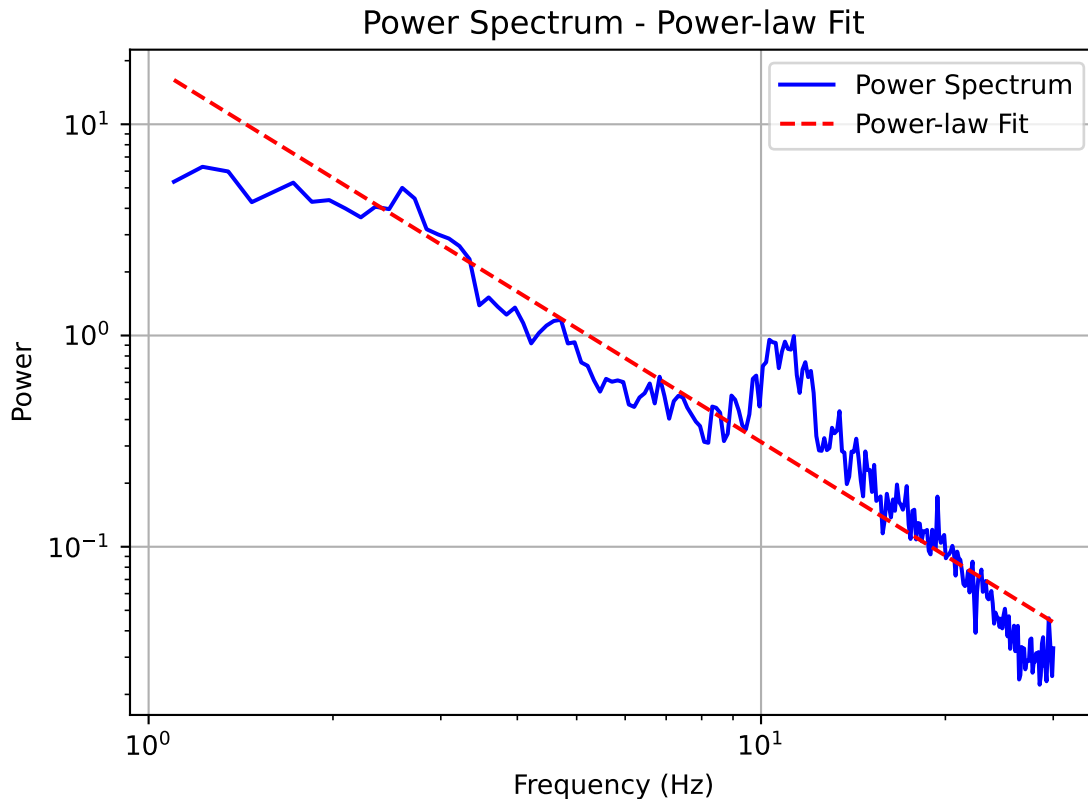
```
plt.ylabel('Power')
```

```
plt.title('Power Spectrum - Power-law Fit')
```

```
plt.legend()
```

```
plt.grid(True)
```

```
plt.show()
```



```
print("Power-law exponent (slope):", slope)
```

```
## Power-law exponent (slope): -1.7878235925781982
```

As shown in the image above, there seems to be a power-law fit. Note that the frequencies and power have been log-transformed. It indicates that the distribution of events or quantities follows a power-law relationship, where the probability of the magnitude of the power in a frequency bin is inversely proportional to its decibel raised to a power. In other words, small frequencies are more common, while larger events or quantities occur less frequently. Since the data is following a power-law fit, we can say it most likely has Scale-free dynamics.

Scale-free dynamics have several important implications, namely:

There is an absence of characteristic scale: Scale-free dynamics suggest that the system does not possess a characteristic scale or time frame that dominates its behavior. Instead, events or quantities can occur at all scales, from small to large.

Self-similarity: Scale-free systems exhibit self-similarity, meaning that the patterns or structures observed at one scale are similar to those observed at other scales. This property is known as fractal behavior.

Emergence of complex behavior: Scale-free dynamics can lead to the emergence of complex behavior and phenomena. The interactions and collective dynamics of the system's components give rise to emergent properties that cannot be easily understood or predicted based solely on the behavior of individual components.

In order to further investigate the Scale-free dynamics, we can use methods such as Fractal analysis and Multifractal analysis.

Periodic and Aperiodic activity

If you look at around the 10-12Hz (within the alpha frequency range), you might notice that there is an increase (that goes quite far above the power-law fit). According to Donoghue et. al., (2020), the power-law behavior can be broken down into two parts. Periodic (or oscillatory) and aperiodic (or 1/f-like). This increase above the power-law fit, is the oscillatory part. We won't dive deeper into this distinction between periodic and aperiodic activity, as it falls outside the scope of this module, however, it is important to know this distinction when reading about further literature regarding to Scale-free activity.

Detrended Fluctuation Analysis

Fractals are mathematical objects that exhibit self-similarity, meaning they display similar patterns at different scales. In the context of time series data, fractal analysis aims to measure the degree of self-similarity or scaling properties within the data (Pilgrim & Taylor., 2019). Fractality can also be observed in brain signals, particularly in the temporal structure of the EEG signal (Stam, C. J 2005). This means that if you zoom in or out on the EEG signal, you may observe similar patterns repeating at different levels of detail. The fractal nature of EEG signals suggests that the brain's activity exhibits a form of temporal self-similarity, where patterns of neural activity repeat across different time scales. This self-similarity can be seen by oscillatory rhythms.

DFA, or Detrended Fluctuation Analysis (Peng et al., 1994), is a mathematical method commonly used to assess the presence of long-range correlations and fractal properties in EEG data (Stam, C. J 2005). It involves dividing the data into smaller segments, known as windows, and fitting a polynomial trend to each segment. By removing the trend from each segment, DFA focuses on the fluctuations or variations around the trend. The root-mean-square fluctuations of these detrended segments are then calculated for different window sizes. The relationship between the window size and the root-mean-square fluctuations determines the scaling properties of the data.

Before continuing, we need to verify that our data is mean-centered.

```
data = epochs.get_data(picks = "eeg")[4][4]
```

```
fig = plt.figure()
plt.plot(data)
```

```
# Add labels and title
```

```
plt.xlabel('Time')
```

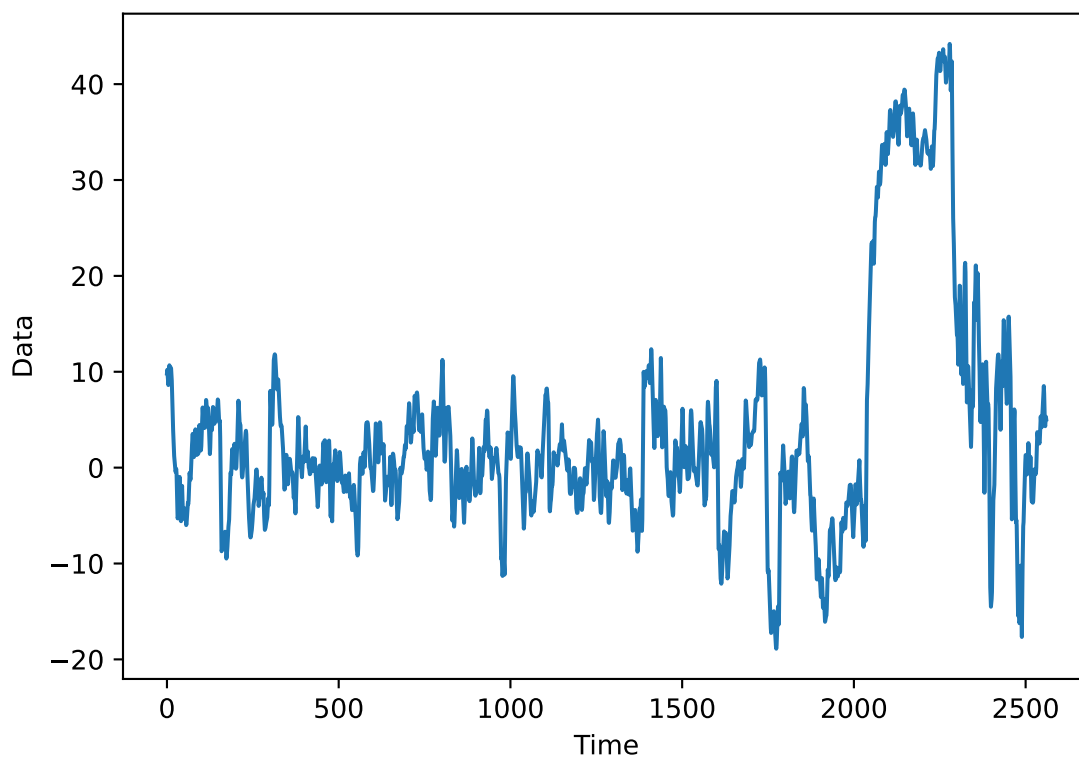
```
plt.ylabel('Data')
```

```
plt.title('The data is not mean centered')
```

```
# Display the plot
```

```
plt.show()
```

The data is not mean centered



```
def is_mean_centered(data):
    mean_diff = np.abs(np.mean(data))
    return mean_diff < 2 #(1% threshold with minmax being 20 and -20, so 0.01 * (20 - (-20)) )

# Assuming you have the necessary data stored in a variable named 'data'
for i in range(9):

    for j in range(8):

        data = epochs.get_data(picks="eeg")[i][j]

        # Check if data is mean-centered
        if is_mean_centered(data):
            #print("Data is mean-centered.")
            continue
        else:
            print("Epoch", i, "channel", j, " is not mean-centered.")

## Epoch 2 channel 4   is not mean-centered.
## Epoch 2 channel 7   is not mean-centered.
## Epoch 3 channel 4   is not mean-centered.
## Epoch 3 channel 7   is not mean-centered.
## Epoch 4 channel 2   is not mean-centered.
## Epoch 4 channel 4   is not mean-centered.
## Epoch 4 channel 5   is not mean-centered.
```

```
## Epoch 4 channel 7 is not mean-centered.
## Epoch 5 channel 2 is not mean-centered.
## Epoch 5 channel 4 is not mean-centered.
## Epoch 5 channel 5 is not mean-centered.
## Epoch 5 channel 7 is not mean-centered.
## Epoch 6 channel 4 is not mean-centered.
## Epoch 6 channel 5 is not mean-centered.
## Epoch 6 channel 6 is not mean-centered.
## Epoch 6 channel 7 is not mean-centered.
## Epoch 7 channel 0 is not mean-centered.
## Epoch 7 channel 3 is not mean-centered.
## Epoch 7 channel 7 is not mean-centered.
```

Here we can see that not all the data is mean centered. We can mean center the data as follows:

```
def mean_center_epochs(epochs):
    # Iterate over all epochs and channels
    for i in range(epochs.get_data().shape[0]):
        for j in range(epochs.get_data().shape[1]):
            data = epochs.get_data()[i, j]
            mean = np.mean(data)
            epochs._data[i, j] -= mean # Subtract the mean from the data
    return epochs
```

```
mean_centered_epochs = mean_center_epochs(epochs)
```

```
require(fractalRegression) # (Likens A, Wiltshire T 2023)
```

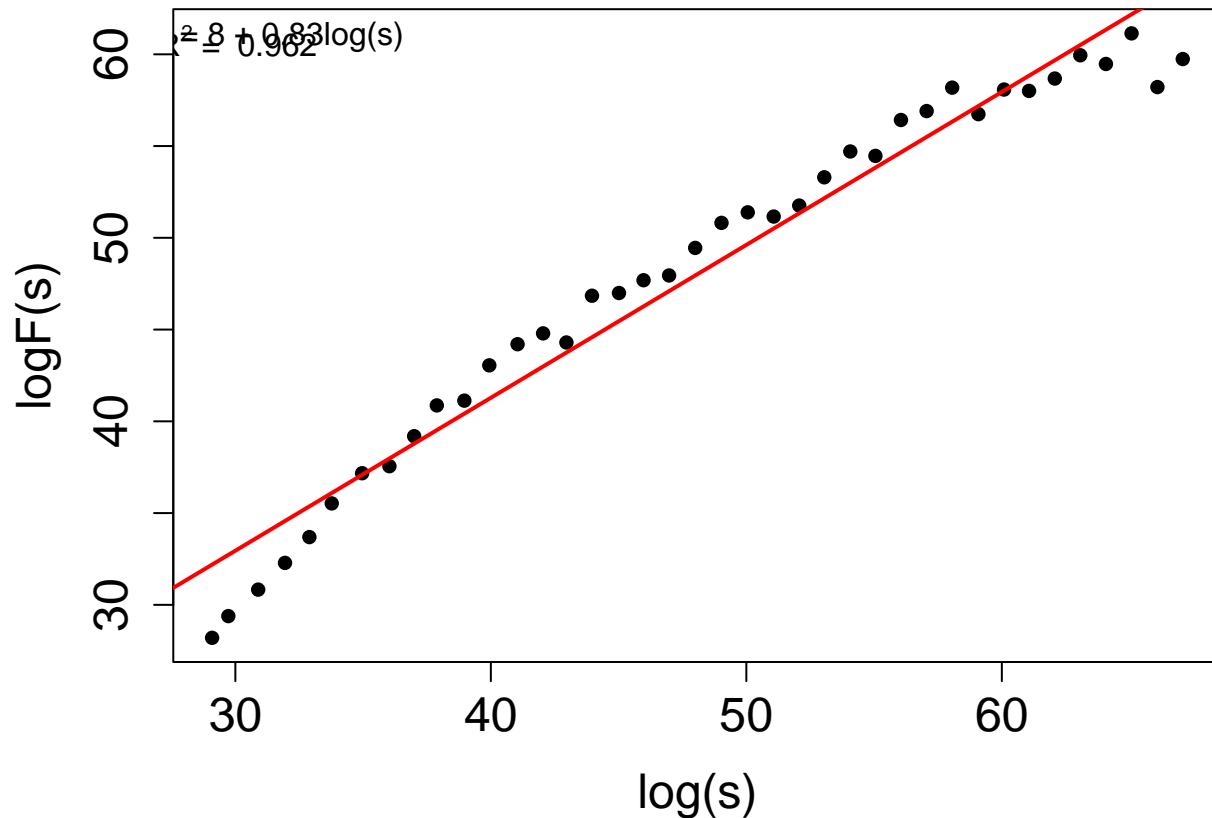
```
epochs <- py$mean_centered_epochs
timeseriesA <- RawEEGComplexSystemsRReadyAdaptiveR[[3]][[1]][["Fz"]]
print(head(timeseriesA))
```

```
0          1          2          3          4          5
-17.54595 -17.77184 -19.44450 -22.60889 -25.39123 -26.62429
```

```
scalesA <- logscale(scale_min = 16, scale_max = length(timeseriesA)/4, scale_ratio = 1.1)
DfaEpoch1Fz <- dfa(x = timeseriesA, order = 1, verbose = 1, scales=scalesA, scale_ratio = 1.1)
cat("Hurst exponent: ", DfaEpoch1Fz$alpha)
```

```
Hurst exponent: 0.8332143
```

```
dfa.plot(DfaEpoch1Fz)
```



The hurst exponent or alpha that is extracted in DFA says something about the scaling properties and long range correlations in a time series. The larger the Hurst exponent is, the stronger the long-range dependencies are within the signal. While it is hard to interpret this number without context, theoretically it could be used in machine learning, or be compared between conditions, as a higher hurst exponent could be an indicator for enhanced sustained attention. In general the signal would be more “persistent” with higher hurst components.

Here we have created a small plot, which loops over the all the epochs in the condition, and plots the Hurst exponent. The epochs range from 1 (near the start of the experiment) to 9 (near the end of the experiment). In this case we see that the Hurst exponent ranges from 0.70 to 0.87, indicating quite some variability. As this is only 1 participant, no conclusions can be drawn from this, but perhaps a preliminary hypothesis could be that around epoch 6 or 7 there is a change in mental vigilance.

```
DfaEpochsFz <- list()
for (i in 1:9){
  timeseriesA <- RawEEGComplexSystemsRReadyAdaptiveR[[i]][[1]][["Fz"]]
  scalesA <- logscale(scale_min = 16, scale_max = length(timeseriesA)/4, scale_ratio = 1.1)
  DfaEpoch <- dfa(x = timeseriesA, order = 1, verbose = 1, scales=scalesA, scale_ratio = 1.1)
  DfaEpochsFz[i] <- DfaEpoch$alpha}

x <- 1:length(unlist(DfaEpochsFz))

# Plot the values
plot(x, unlist(DfaEpochsFz), type = "o", pch = 16, col = "blue", xlab = "Index", ylab = "Value", main =
```



Multifractal Detrended Fluctuation Analysis

An addition onto DFA is Multifractal Detrended Fluctuation Analysis (MFDFA). In the case of MFDFA, the method is extended to capture the multifractal nature of the time series (Keltz-Stephen et al., 2016). Multifractals are a type of fractal system where the scaling properties vary across the data, indicating the presence of multiple fractal dimensions (Papo et al., 2017). This means that different parts of the data exhibit different degrees of self-similarity. Where only the global scaling component is used in DFA, in MFDFA the local scaling component is also utilized.

Before continuing, we should verify that our data indeed shows multifractal behavior. We can do this by either looking at the Multifractal spectrum, which shows the local scaling components. If the lines on the plot are parallel, the signal does not exhibit multifractal behavior. If the lines are concave or convex the signal does show multifractal behavior.

Note that q values between 1 and 20 are picked here, as we only have 2561 values, the negative q values might give unrealistic outputs (Ihlen & Vereijken, 2010).

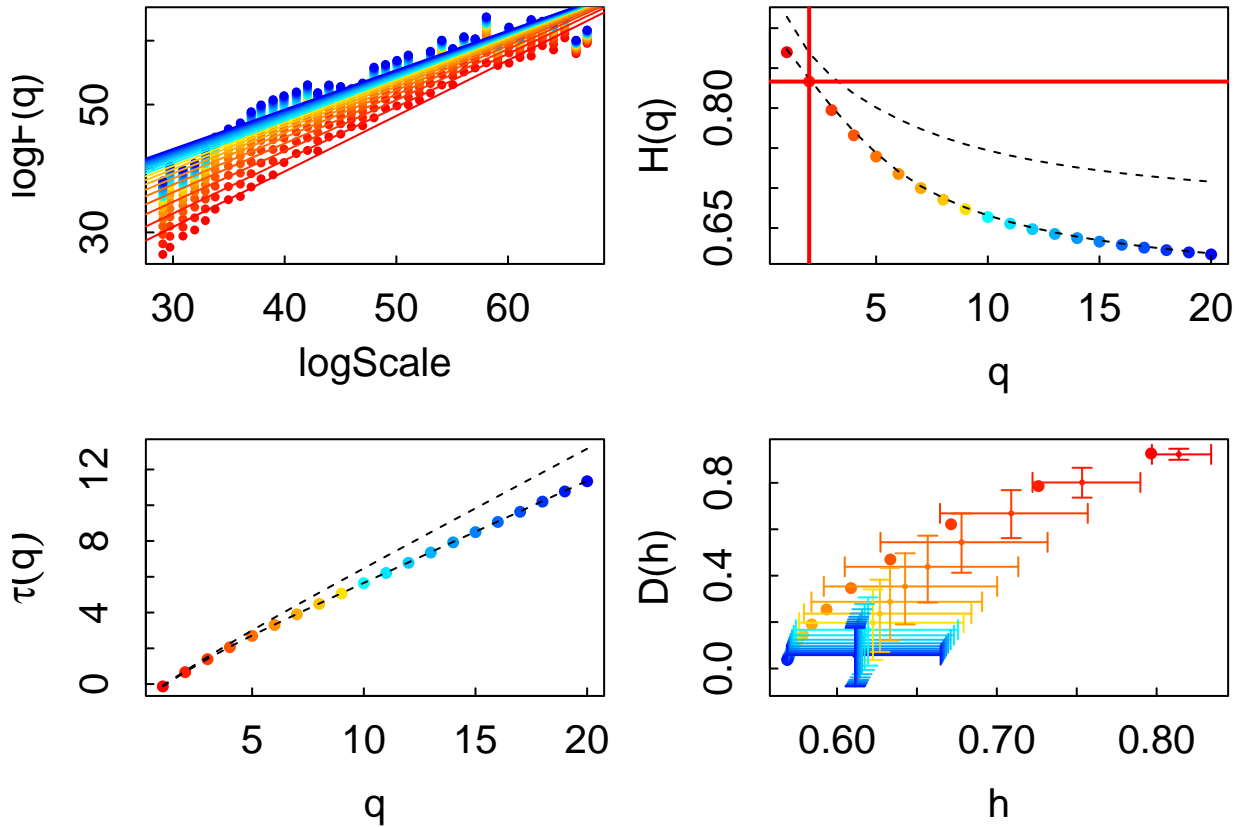
```
timeseriesA <- RawEEGComplexSystemsRReadyAdaptiveR[[3]][[1]][["Fz"]]

scalesA <- logscale(scale_min = 16, scale_max = length(timeseriesA)/4, scale_ratio = 1.1)

timeseriesA.mf.dfa.out <- mfdfa(x = timeseriesA, q = c(1:20), order = 1, scales=scalesA, scale_ratio=1.1)
timeseriesA.surr.mf.dfa.out <- mfdfa(x = sample(timeseriesA, replace=FALSE), q = c(-10:10), order = 1,
```



```
mfdfa.plot(timeseriesA.mf.dfa.out, do.surrogate = TRUE)
```



Q-order fluctuation function

The top-left plot here is the q -order fluctuation function. To see whether we have multifractality in our signal, we look at whether the lines are going inwards, or whether they are going parallel. As we can see, the distance between the lines at the left of the plot are wider than the lines at the right side of the plot, suggesting multifractal behavior. We can get the degree of multifractality by extracting the maximum distance between the lines.

```
Distance <- max(timeseriesA.mf.dfa.out$h) - min(timeseriesA.mf.dfa.out$h)
print(Distance)
```

```
## [1] 0.2276805
```

A wider distance implies that the data contains regions or segments with distinct scaling properties, indicating a complex and heterogeneous system. If the distance is relatively small it suggests a less pronounced multifractal behavior. The signal may exhibit some degree of self-similarity or scaling properties, but with less variation across different scales or positions.

Renyi scaling exponent $t(q)$ over q

In multifractal analysis, the Renyi scaling exponent, denoted as $t(q)$, describes the relationship between q and the scaling exponent t . The scaling exponent $t(q)$ helps us understand how the fluctuations in the time

series scale at different moments. It gives us insights into the varying scaling properties of the data across different scales or levels of detail.

The Renyi scaling exponent $t(q)$ is related to the generalized Hurst exponent $H(q)$ through the equation $t(q) = qH(q)$.

If the plot is curved or is non-linear, it suggests the presence of multifractality. In this case, the plot is slightly curved, suggesting multifractality

q is the moment order, which is a mathematical quantity used to describe the shape, location, and spread of a probability distribution. In the context of multifractality, the moment order determines the specific statistical property or moment that we are interested in examining. The choice of the moment order allows us to focus on different aspects of the data's variability. For example, when $q=2$, we are looking at the second moment, which is the variance of the data. The second moment provides information about the spread or dispersion of the data points. When $q=3$, we are looking at the third moment, which is related to skewness and provides insights into the asymmetry of the distribution. Higher moment orders, such as $q=4$, $q=5$, and so on, capture higher-order statistical properties of the data.

Hurst Exponent

The $H(q)$ over q plot in the top right shows the relationship between q and the corresponding generalized Hurst exponent $H(q)$. The generalized Hurst exponent $H(q)$ characterizes the scaling behavior of the fluctuations at different moment orders in the time series.

This plot shows whether there are long-range correlations and whether scaling is present in the signal.

What is interesting, is that if we look at $q=2$, then we are performing the same analysis as normal dfa. Here we can see that it is the same value as the hurst exponent when we did regular DFA.

```
timeseriesA.mf.dfa.out$Hq[timeseriesA.mf.dfa.out$q==2]
```

```
## [1] 0.8332143
```

Note here that with normal DFA, we are interested in the variance in a signal. Remember that DFA removes the trend, and then looks at the fluctuations at different scales of what is left (the variance). Since $q=2$, is the moment of variance, this is the exact same!

Singularity spectrum

In this plot on the bottom right, the h -axis represents the singularity exponent, which characterizes the local behavior of the time series at different scales. The $D(h)$ values on the vertical axis correspond to the generalized fractal dimensions associated with each singularity exponent. These dimensions quantify the scaling properties and complexity of the data. Each point on the plot corresponds to a specific fractal dimension associated with a singularity exponent. It provides information about the range and strength of fractal dimensions present in the data. When the shape is curved or non-linear, it suggests multifractality. In other words it mean that different parts of the eeg data exhibit different scaling behaviors and complexities

The singularity exponent should not get confused with the Hurst exponent (in the $H(q)$ over q plot)

Investigating multifractal spectrum width in different brain regions

Here we investigate the multifractal spectrum width over all participants and epochs, and average the distance. By doing so, we can get the average distance within each electrode, which can give us a fairly accurate indication of the amount of multifractality within the underlying brain regions.

The code is not included as it is a massive piece of code. The code loops over the participants, epochs, and electrodes and calculates the multifractal spectrum for each electrode. It also calculates this for a surrogate of the data. The following variables were used in the mfdfa analysis.

```
scalesA <- logscale(scale_min = 16, scale_max = length(timeseriesA)/4, scale_ratio = 1.1)
timeseriesA.mf.dfa.out <- mfdfa(x = timeseriesA, q = c(1:10), order = 1, scales=scalesA, scale_ratio=1.1)
```

```
# Calculate the average of Fz_listMFDFA
avg_Fz <- mean(Fz_listMFDFA)

# Calculate the average of C3_listMFDFA
avg_C3 <- mean(C3_listMFDFA)

# Calculate the average of Cz_listMFDFA
avg_Cz <- mean(Cz_listMFDFA)

# Calculate the average of C4_listMFDFA
avg_C4 <- mean(C4_listMFDFA)

# Calculate the average of Pz_listMFDFA
avg_Pz <- mean(Pz_listMFDFA)

# Calculate the average of P07_listMFDFA
avg_P07 <- mean(P07_listMFDFA)

# Calculate the average of Oz_listMFDFA
avg_Oz <- mean(Oz_listMFDFA)

# Calculate the average of P08_listMFDFA
avg_P08 <- mean(P08_listMFDFA)

cat("Average of Fz_listMFDFA:", avg_Fz, "\n")
```

```
## Average of Fz_listMFDFA: 0.1987257
```

```
cat("Average of C3_listMFDFA:", avg_C3, "\n")
```

```
## Average of C3_listMFDFA: 0.1729857
```

```
cat("Average of Cz_listMFDFA:", avg_Cz, "\n")
```

```
## Average of Cz_listMFDFA: 0.1824905
```

```
cat("Average of C4_listMFDFA:", avg_C4, "\n")
```

```
## Average of C4_listMFDFA: 0.1822479
```

```
cat("Average of Pz_listMFDFA:", avg_Pz, "\n")
```

```
## Average of Pz_listMFDFA: 0.1871776
```

```
cat("Average of P07_listMFDFA:", avg_P07, "\n")
```

```
## Average of P07_listMFDFA: 0.1747223
```

```
cat("Average of Oz_listMFDFA:", avg_Oz, "\n")
```

```
## Average of Oz_listMFDFA: 0.1795113
```

```
cat("Average of P08_listMFDFA:", avg_P08, "\n")
```

```
## Average of P08_listMFDFA: 0.1800182
```

The Fz electrode has the highest average value. It is positioned over the frontal area of the brain, which is involved in executive functions, attention, and decision-making. These cognitive processes may exhibit more complex and multifractal dynamics. It might also be the case that, since the participants had to pay attention to the task, this area of the brain was very active, and had more underlying processes. These findings could be in line with Olejarczyk et al., (2021), who found higher complexity in the frontal lobe in sleeping participants.

Another observation could be that posterior regions have higher spectrum width than Central regions. The differences in multifractal width values may be related to the sensory modality being recorded. For example, the posterior electrodes often capture visual or auditory information, which can have unique dynamics and scaling properties compared to the other electrodes involved in higher-order cognitive functions. The sensory-specific processing and neural interactions in these regions may contribute to variations in multifractal behavior.

Comparing to Surrogate data

During the analysis we also added surrogate data. With a surrogate analysis we take the original signal and randomize/shuffle itself. This way we can see whether the data truly shows fractality based on how the signal is structured. If the signal is not significantly different from a randomly shuffled version of itself, it might mean that there isn't any fractality going on. Here we use the distance measure as a way to show whether the Surrogate data is significantly different from the original EEG data.

To do this we use a paired Wilcoxon signed-rank test (a non-parametric test) (as it violates assumptions for a regular t-test). We also do Bonferroni correction, as multiple channels are compared at once.

```
# Perform Wilcoxon signed-rank test for each channel
channel_list <- c("Fz", "C3", "Cz", "C4", "Pz", "P07", "Oz", "P08")
alpha <- 0.05 # Significance level

for (channel in channel_list) {
  channel_data_adaptive <- get(paste0(channel, "_listAMFDFA"))
  channel_data_adaptive_surg <- get(paste0(channel, "_listAMFDFA_Surg"))

  # Perform Wilcoxon signed-rank test
  wilcox_result <- wilcox.test(channel_data_adaptive, channel_data_adaptive_surg, paired = TRUE)

  # Adjust p-value with Bonferroni correction
  p_value_adjusted <- wilcox_result$p.value * length(channel_list)
```

```

# Determine significance based on adjusted p-value
is_significant <- p_value_adjusted < alpha

# Print the results
cat("Channel:", channel, "\n")
cat("Adjusted p-value:", p_value_adjusted, "\n")
cat("Significant:", is_significant, "\n")
cat("Median Adaptive:", median(channel_data_adaptive), "\n")
cat("Median Adaptive Surrogate:", median(channel_data_adaptive_surg), "\n")
cat("\n")
}

```

```

## Channel: Fz
## Adjusted p-value: 2.490001e-13
## Significant: TRUE
## Median Adaptive: 0.1948478
## Median Adaptive Surrogate: 0.2313377
##
## Channel: C3
## Adjusted p-value: 4.020887e-19
## Significant: TRUE
## Median Adaptive: 0.1726118
## Median Adaptive Surrogate: 0.2232064
##
## Channel: Cz
## Adjusted p-value: 4.136156e-14
## Significant: TRUE
## Median Adaptive: 0.1819735
## Median Adaptive Surrogate: 0.2207213
##
## Channel: C4
## Adjusted p-value: 4.019409e-23
## Significant: TRUE
## Median Adaptive: 0.170974
## Median Adaptive Surrogate: 0.2256752
##
## Channel: Pz
## Adjusted p-value: 8.157787e-17
## Significant: TRUE
## Median Adaptive: 0.1841106
## Median Adaptive Surrogate: 0.2291066
##
## Channel: P07
## Adjusted p-value: 5.646429e-18
## Significant: TRUE
## Median Adaptive: 0.1684702
## Median Adaptive Surrogate: 0.2283917
##
## Channel: Oz
## Adjusted p-value: 6.28766e-20
## Significant: TRUE
## Median Adaptive: 0.1743847
## Median Adaptive Surrogate: 0.2220452

```

```
##  
## Channel: P08  
## Adjusted p-value: 6.816307e-20  
## Significant: TRUE  
## Median Adaptive: 0.16732  
## Median Adaptive Surrogate: 0.2247501
```

As we can see here, it is indeed the case that the EEG-based distances are significantly different from the Surrogate-based distances, once again suggesting multifractality.

Conclusion

In conclusion, our investigation in this notebook provided significant insights into the nature of EEG data. Firstly, we established that the data follows a scale-free or power-law behavior, indicating a complex underlying structure. This finding suggests that EEG signals exhibit long-range correlations and self-similarity across different time scales.

We then employed Detrended Fluctuation Analysis (DFA) to delve deeper into the characteristics of the EEG data. DFA allowed us to assess the presence of fractal patterns and quantify the correlation properties of the signals. The analysis further confirmed the scale-free behavior observed earlier and provided additional evidence for the self-similar nature of the EEG data.

Moreover, by employing multiple analysis techniques, we unveiled that the EEG data exhibit multifractal behavior. This implies that the data possesses a wide range of statistical properties, varying across different scales. The presence of multifractality suggests that EEG signals have complex and heterogeneous dynamics, characterized by variations in both local and global scaling properties.

Overall, our findings highlight the rich and intricate nature of EEG data, shedding light on its multifractal and self-similar properties. These results contribute to a deeper understanding of the underlying mechanisms governing brain activity and provide a foundation for further research and analysis in the field of EEG signal processing and characterization.

References

- Bénar CG, Grova C, Jirsa VK, Lina JM. Differences in MEG and EEG power-law scaling explained by a coupling between spatial coherence and frequency: a simulation study. *J Comput Neurosci*. 2019 Aug;47(1):31-41. doi: 10.1007/s10827-019-00721-9. Epub 2019 Jul 11. PMID: 31292816.
- Bansal, K., Garcia, J. O., Lauharatanahirun, N., Muldoon, S. F., Sajda, P., & Vettel, J. M. (2021). Scale-specific dynamics of high-amplitude bursts in EEG capture behaviorally meaningful variability. *NeuroImage*, 241, 118425.
- Donoghue T, Haller M, Peterson EJ, Varma P, Sebastian P, Gao R, Noto T, Lara AH, Wallis JD, Knight RT, Shestyuk A, Voytek B. Parameterizing neural power spectra into periodic and aperiodic components. *Nat Neurosci*. 2020 Dec;23(12):1655-1665. doi: 10.1038/s41593-020-00744-x. Epub 2020 Nov 23. PMID: 33230329; PMCID: PMC8106550.
- Ihlen, E. A., & Vereijken, B. (2010). Interaction-dominant dynamics in human cognition: Beyond 1/f α fluctuation. *Journal of Experimental Psychology: General*, 139(3), 436.
- Pilgrim, I., & P. Taylor, R. (2019). Fractal Analysis of Time-Series Data Sets: Methods and Challenges. *IntechOpen*. doi: 10.5772/intechopen.81958
- Peng, C-K., et al. "Mosaic organization of DNA nucleotides." *Physical review e* 49.2 (1994): 1685.
- Papo, David; Goñi, Joaquin; Buldú, Javier M. (2017). "Editorial: On the relation of dynamics and structure in brain networks".

Stam, C. J. (2005). Nonlinear dynamical analysis of EEG and MEG: review of an emerging field. *Clinical neurophysiology*, 116(10), 2266-2301.

Kelty-Stephen, D. G., Stirling, L. A., & Lipsitz, L. A. (2016). Multifractal temporal correlations in circle-tracing behaviors are associated with the executive function of rule-switching assessed by the Trail Making Test. *Psychological assessment*, 28(2), 171.

Olejarczyk, E., Gotman, J. & Frauscher, B. Region-specific complexity of the intracranial EEG in the sleeping human brain. *Sci Rep* **12**, 451 (2022). <https://doi.org/10.1038/s41598-021-04213-8>

Garcia C (2022). `_nonlinearTseries: Nonlinear Time Series Analysis_`. R package version 0.2.12, https://CRAN.R-project.org/package=_nonlinearTseries

Antonio, Narzo FD (2019). `_tseriesChaos: Analysis of Nonlinear Time Series_`. R package version 0.1-13.1, https://CRAN.R-project.org/package=_tseriesChaos

Ushey K, Allaire J, Tang Y (2023). `_reticulate: Interface to 'Python'_`. R package version 1.28, https://CRAN.R-project.org/package=_reticulate

Likens A, Wiltshire T (2023). `_fractalRegression: Performs Fractal Analysis and Fractal Regression_`. R package version 0.1-1, https://CRAN.R-project.org/package=_fractalRegression

H. Wickham. `ggplot2: Elegant Graphics for Data Analysis`. Springer-Verlag New York, 2016.

Borchers H (2022). `_pracma: Practical Numerical Math Functions_`. R package version 2.4.2, https://CRAN.R-project.org/package=_pracma